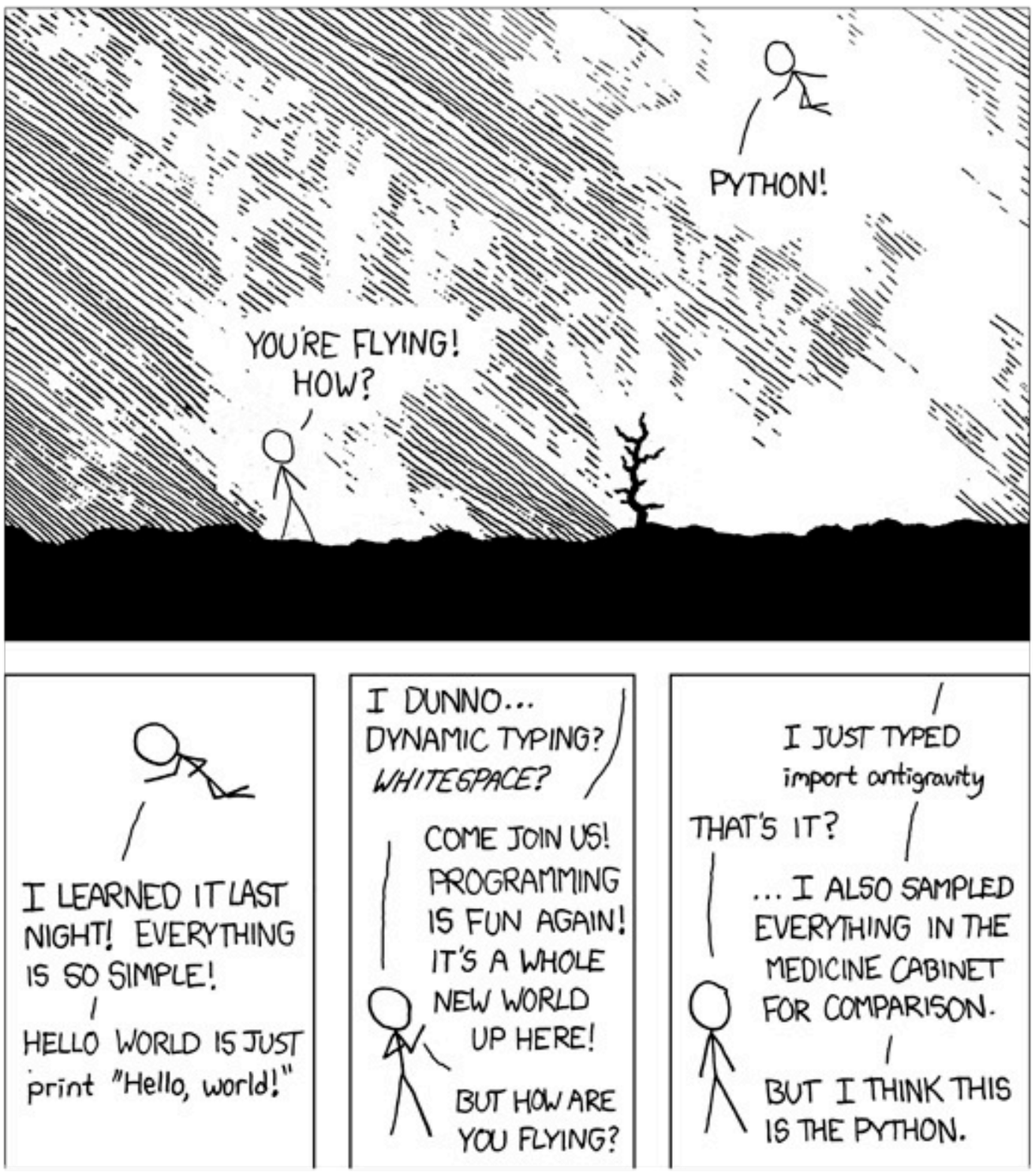


http://xkcd.com/353/

Internet Programming in Python

Week 1 Networking

Brian Dorsey
brian@dorseys.org

Overview & Introductions

(25)

Magic

Magic

(or at least, indistinguishable from)

This is our goal. To make things which are useful and save time, yes. They should also delight people. Surprise them by doing just a bit more than expected.

Embarrassment of riches

We’re all standing on the shoulders of giants here.
Each of you has the power to create amazing new things... as a hobby!
With this leverage, even a couple of hours a week is amazing.

Beware the trappings of authority.

I’m not the smartest one in the room. But this room is designed to make it look and feel that way. Who designed it? You’re trapped. Apologies. This is what our culture thinks ‘learning’ looks like.
One thing we can do, is make this more like a discussion than a lecture - ask questions anytime.

Collaboration

Real learning happens when doing.
Real learning happens when teaching.
Software is crazy specialized, lets all help each other get better at it.

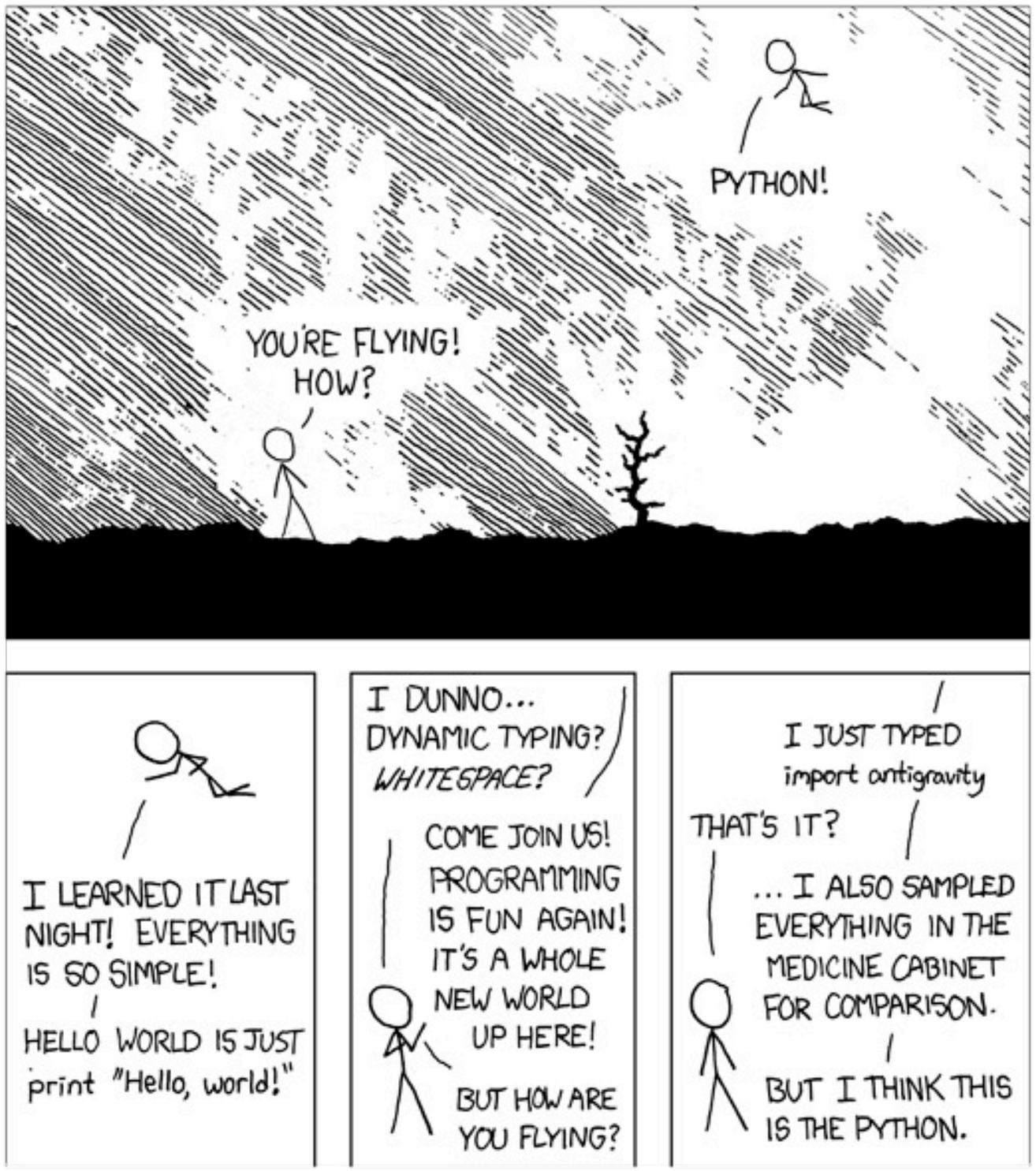
"One of the things I find that's remarkable about Python is that it has a very even learning curve. Maybe it's not even a curve, It's kind of a straight line."

- Bruce Eckel

http://www.artima.com/intv/prodperfP.html

(yes, this is an appeal to authority... two sides after cautioning against it.)

(imagine graph here)



http://xkcd.com/353/

The flat learning curve leads us here.

http://commons.wikimedia.org/wiki/File:Buffet-2.jpg

Think of this class as the first pass through a buffet line. Theres an amazing assortment of food, but our plates are only so big. We're going to grab a bite of everything, then sit down and taste each one. By the end of the course, you'll know what you want to go back and dig into.

Abstraction

Nearly every class we're going to add a level of abstraction.
By the end, Google will be doing everything for us. :)

Logistics

Virtual Machines

- Donation from Blue Box Group
- 3 months * 25 machines = approximately $10,000 donation
- If you, or someone you know needs hands-on hosting, talk to them.

Virtual Machines

- Ubuntu Linux 10.4 (Lucid)
- Python 2.6.5
- 512 MB RAM
- 10 GB storage
- Reliable internet connection
- On 24/7 - until the end of the course!

Also, we need to support each other, don't contact Blue Box Group for support. Try not to go over on RAM or storage. Don't download the whole internet.

Dropbox

UWIDs

I’m going to use your UWIDs as a ‘primary key’ of sorts. Specifically for assignments. Sometimes just turning things in, sometimes as part of the services we write.

Labs

Group work - as well as we can, given the constraints of this room.
4ish people if possible, pairs otherwise.
Per lab volunteers to help. We've got a wide range of skills, if you already know a particular topic (networking, Django, etc0, help others.
Rule of thumb: if you get bored during a lab, stand up and start teaching, you'll start learning again. :)

Assignments

Doing assignments in groups is absolutely OK. You'll learn more and better.
(if that sounds like a way out of doing the assignments, you're better off dropping this course now)
Today's assignment is a getting-started one - we'll have time today to get some of the infrastructure stuff setup.

Introductions (again?)

Lecture A (20)

Networking:
from hardware to sockets

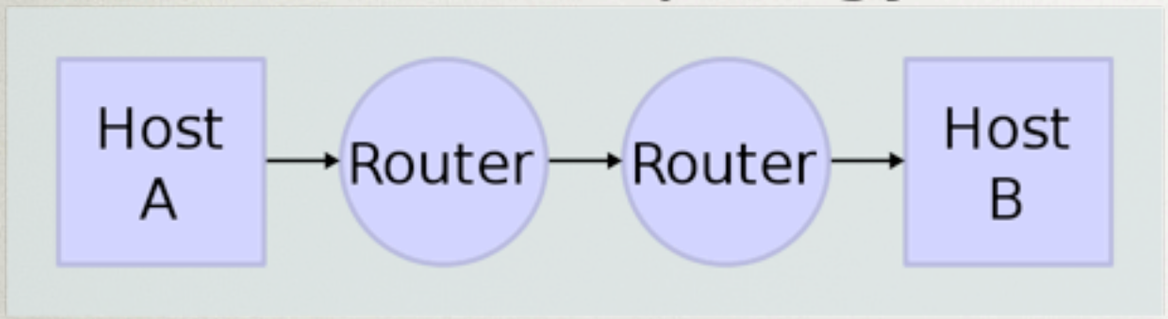
level setting

- domain names (DNS)
- IP address
- port
- tcp
- udp

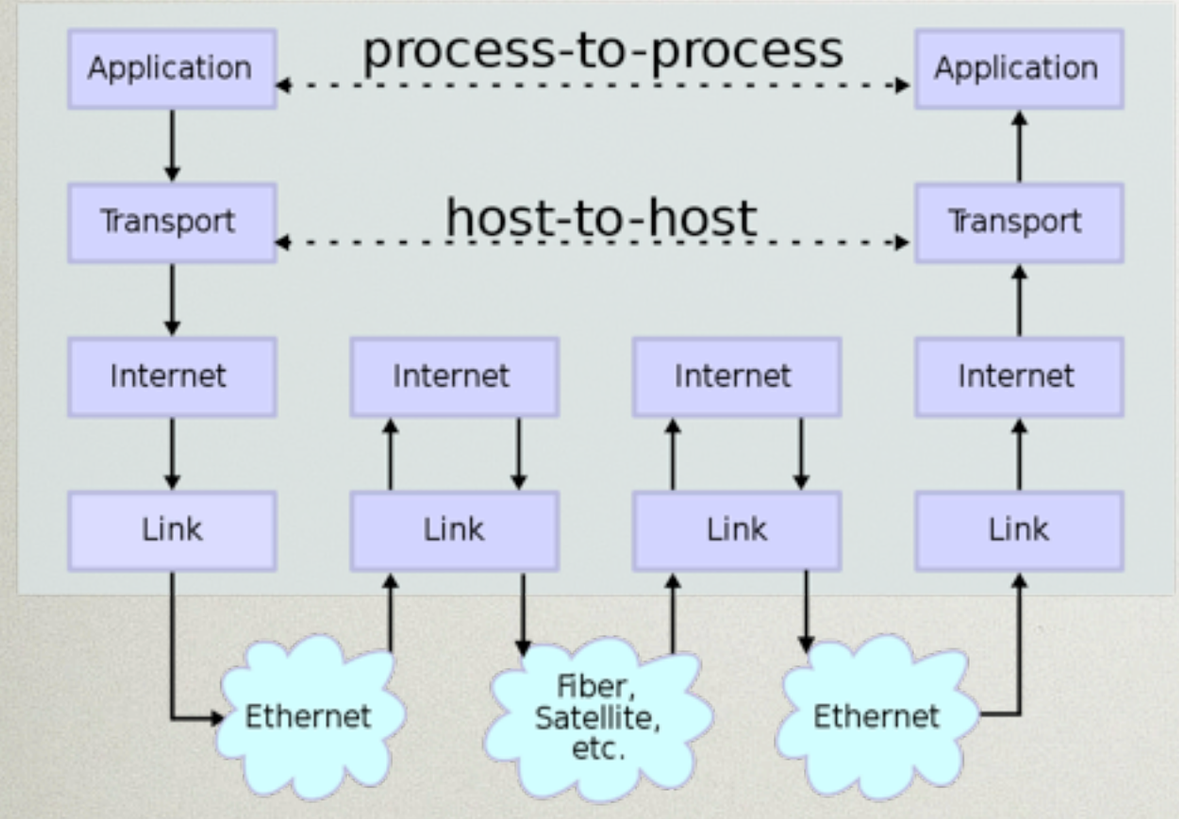
questions from the readings

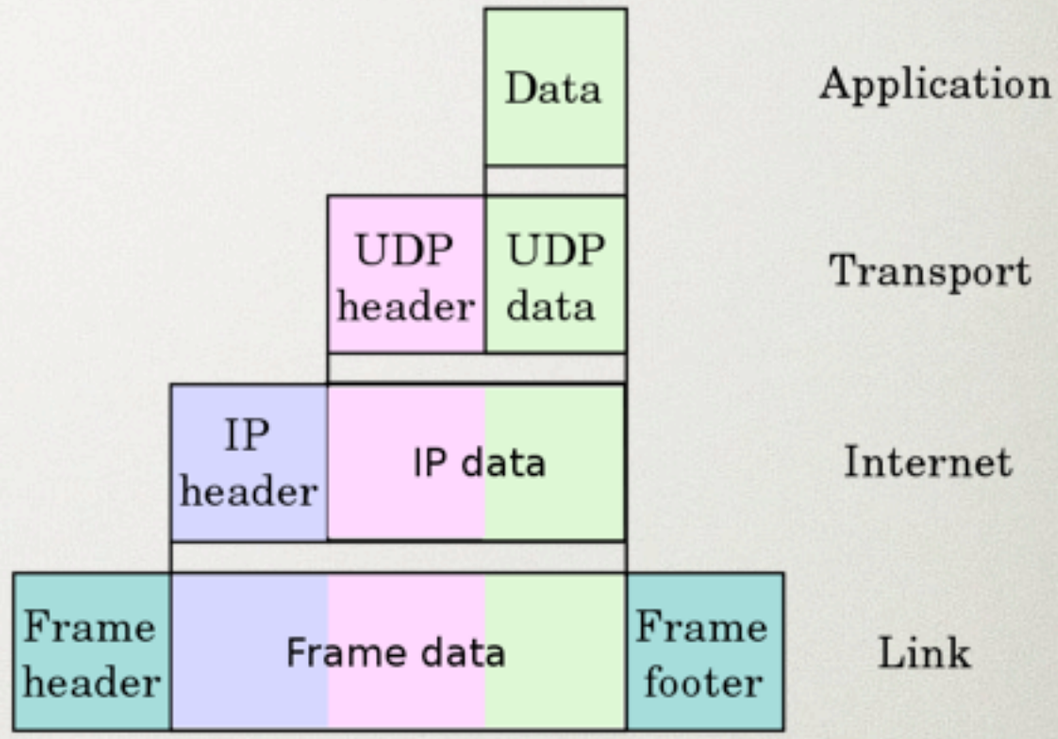
How was the reading?
Questions?

Network Topology



Data Flow





http://en.wikipedia.org/wiki/Internet_Protocol_Suite

We're only going to talk about half of this. ;)

IP addresses

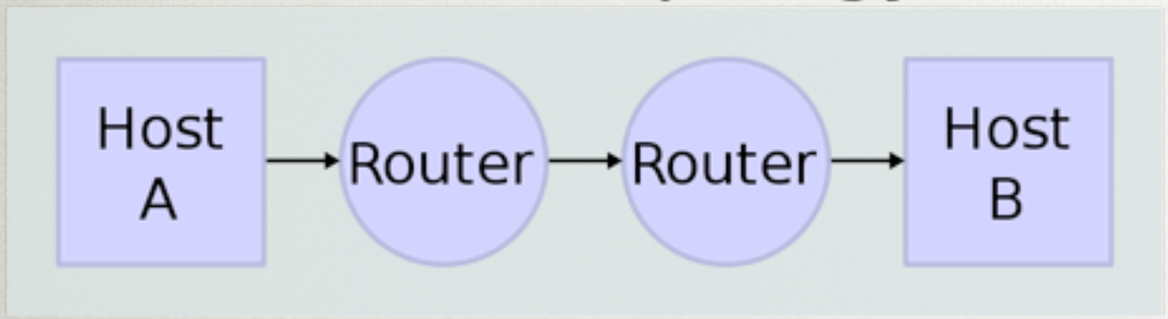
ports

25, 80

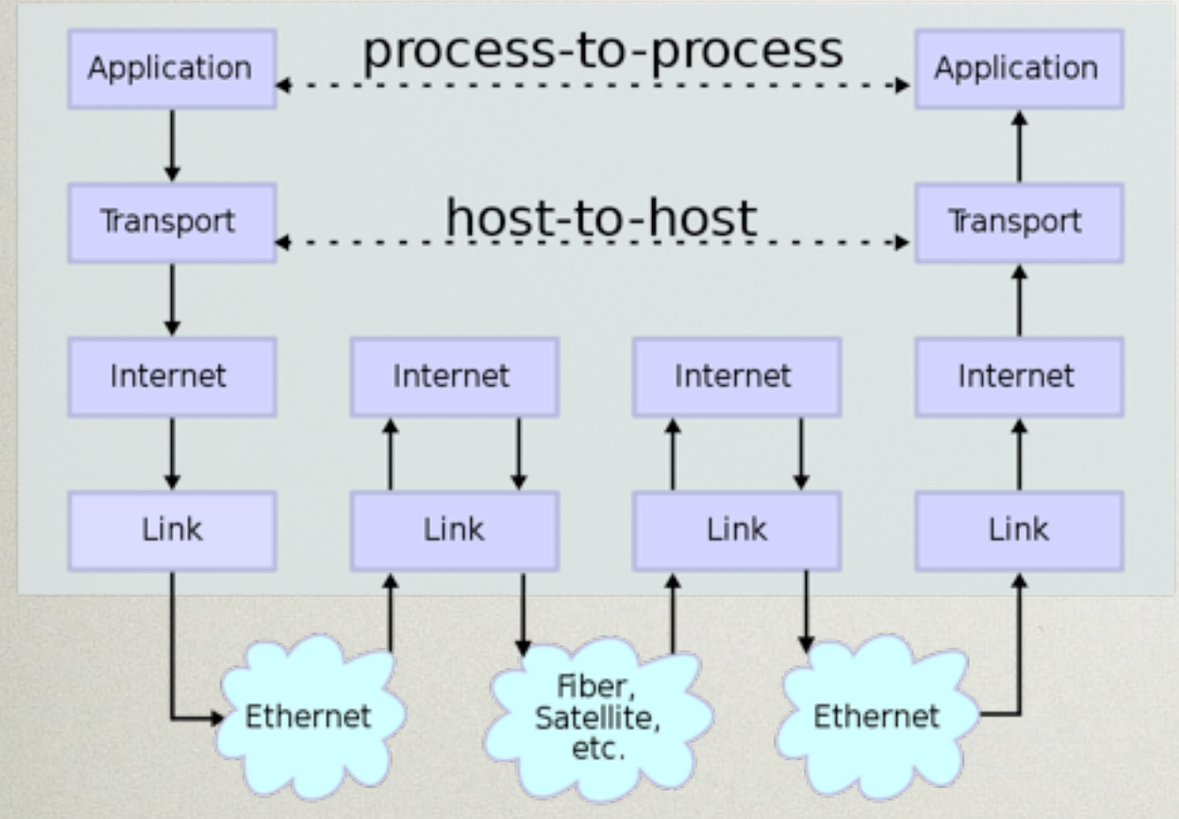
transports

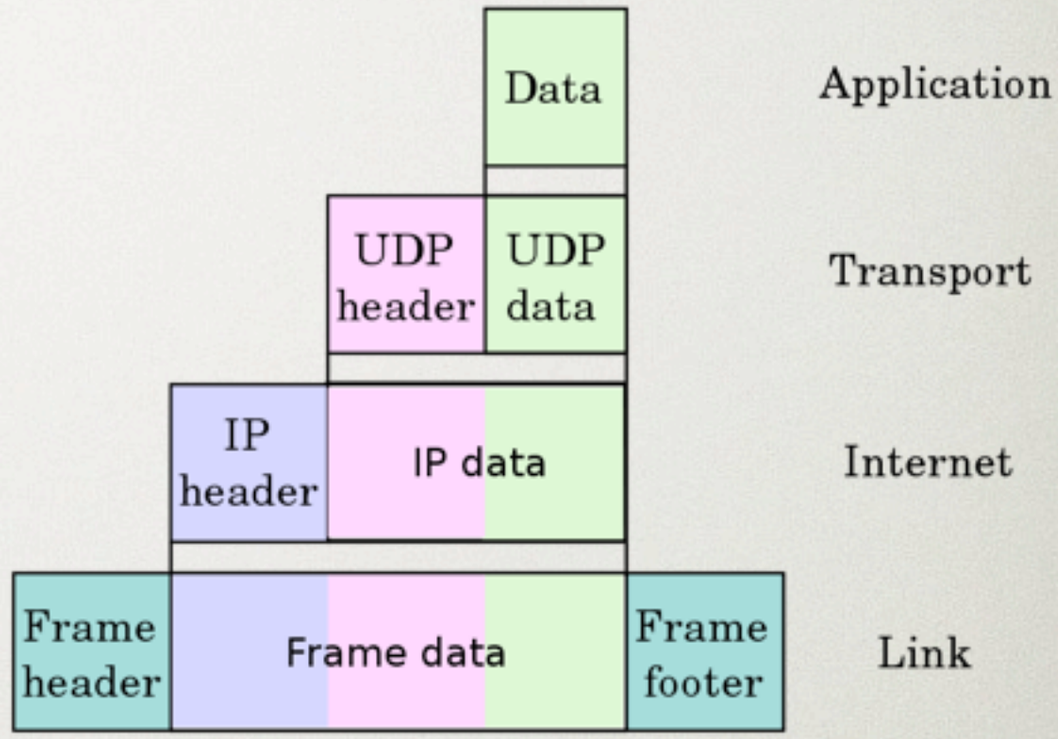
TCP, UDP

Network Topology



Data Flow





http://en.wikipedia.org/wiki/Internet_Protocol_Suite

We're only going to talk about half of this. ;)
Nearly the entire class will be at the application level, and most of that in protocols built on top of HTTP.

Socket abstraction
(IP, port, transport)

text vs binary

Technically, it's all binary, but lots of protocols limit themselves to plain text.

(echo server)

```
import socket

host = ''
port = 50000
backlog = 5
size = 1024
s = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM)
s.bind((host,port))
s.listen(backlog)
while True:
    client, address = s.accept()
    data = client.recv(size)
    if data:
        client.send(data)
    client.close()
```

http://ilab.cs.byu.edu/python/socket/echoserver.html

(echo client)

```
import socket

host = 'localhost'
port = 50000
size = 1024
s = socket.socket(socket.AF_INET,
                  socket.SOCK_STREAM)
s.connect((host,port))
s.send('Hello, world')
data = s.recv(size)
s.close()
print 'Received:', data
```

http://ilab.cs.byu.edu/python/socket/echoclient.html

Lab A
(20)

groups of 4ish if possible, pairs if not

Lab A

- Write a program which connects via TCP to (IP, port) and prints everything it receives.
- Bonus: Make your own implementation of the server.
- IP: block115379-pwc.blueboxgrid.com port: 50000

Break
(10)

Lecture B (20)

Dig into a protocol: Simple Mail Transport Protocol (SMTP)

Simple Mail Transfer Protocol (SMTP)

Nearly every email is sent via SMTP.
Not used for checking mail (POP3, IMAP)

SMTP

- Connection oriented (TCP)
- request / reply
- status codes & messages

(demo)

```
$ telnet mail.blueboxgrid.com 25

S: 220 mail2.blueboxgrid.com ESMTP Postfix
C: helo  block115379-pwc.blueboxgrid.com
S: 250 mail2.blueboxgrid.com
C: mail from:<brian@dorseys.org>
S: 250 2.1.0 Ok
C: rcpt to:<briandorsey@gmail.com>
S: 250 2.1.5 Ok
C: data
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: This is the subject.

This is the body of the message.
.
S: 250 2.0.0 Ok: queued as 6732C530233
C: quit
S: 221 2.0.0 Bye
```

(evil demo)

Python module: smtplib

Lab B
(20)

Lab B

- Individually log into your VM, telnet to mail.blueboxgrid.com and manually send an email to yourself.
- With your group, write a mail merge script, which sends a customized email to everyone in your group.

Break
(10)

Example Lightning Talk (5)

(OPEN)

Lightning Talks (10)

history

Must include

- what is it?
- why should we care?
- code sample

Who wants to go next week?

Better Example (10)

Assignment (30)

Assignment

- Use sockets to write a server which takes two numbers, adds them together and returns the result.
- Write a client for the above server.

- Turn in here: http://bit.ly/uwipip_week1

Checklist

- make sure you can ssh into VM
- sign up for dropbox and make sure you can see the uwpython shared folder
- sign up for github.com
- install git or svn on your machine
- setup repository for assignments
- check out repository to your dev machine
- do assignment
- turn it in by Monday evening @ 7pm

Working through lots of fiddly setup stuff.

Info

- Midweek gathering:
  Sunday Jan 16, 2-5pm
  Tully's at 4th & Union